日期: 12 / 17

Mini-batch 梯度下降.

$m = 5{,}000{,}000$. 5000个 batch, batch_size = 1000.

$X^{\{t\}}, Y^{\{t\}}$.

(1000个) (1000个). for t=1 to 5000:

对 $X^{\{t\}}, Y^{\{t\}}$ 做梯度下降.

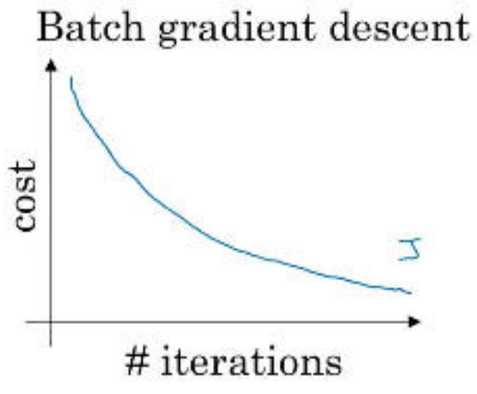
$$\text{cost} = J^{\{t\}} = \frac{1}{1000} \cdot \sum_{i=1}^{1000} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2\times 1000} \|w^{[l]}\|_F$$

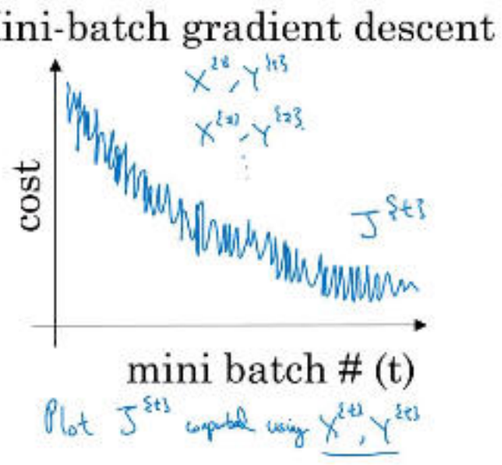
Back Propagate.

Gradient Descent.

一个 epoch: 遍历仅, 了 一次, 已经做了5000次梯度下降.
但每一次都较快.

Training with mini batch gradient descent





Andrew Ng

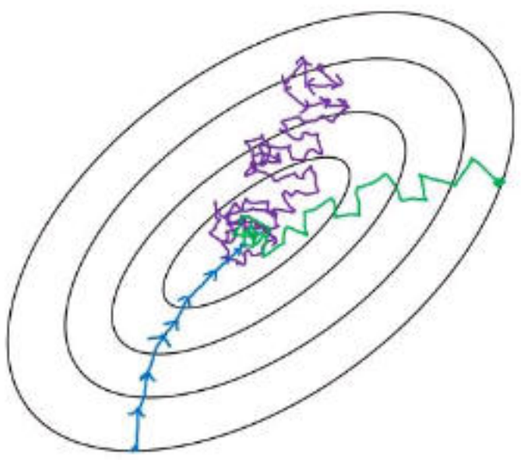
日期: 12 / 17

Choosing your mini-batch size

→ If mini-batch size = m : Batch gradient descent. $(X^{\{1\}}, Y^{\{1\}}) = (X, Y)$.

→ If mini-batch size = 1 : Stochastic gradient descent. Every example is it own $(X^{\{1\}}, Y^{\{1\}}) = (x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$ mini-batch.

In practice: Somewhere in-between 1 and m



Stochastic gradient descent
↓
Lose speedup from vectorization

In-between (mini-batch size not too big/small)
↓
Fastest learning.
- Vectorization. (n 1000)
- Make progress without processing entire training set.

Batch gradient descent (mini-batch size = m)
↓
Too long per iteration

Andrew Ng

(Gradient Descend).

batch_size = 1: 一个样本做一次梯度下降, ⇒ 随机 GD.

batch size = m: 所有样本, 全局优化, 可能很慢 (m大)

batch size 常取 $2^6, \dots, 2^{10}$. 2进制, ↓

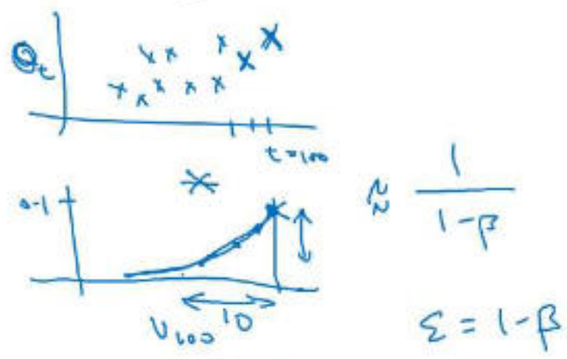
但方向性强.

日期: 12/17

指数加权平均.

Exponentially weighted averages

$v_t = \beta v_{t-1} + (1-\beta)\theta_t$



$\approx \frac{1}{1-\beta}$

$\varepsilon = 1-\beta$

$v_{100} = 0.9v_{99} + 0.1\theta_{100}$

$v_{99} = 0.9v_{98} + 0.1\theta_{99}$

$v_{98} = 0.9v_{97} + 0.1\theta_{98}$

...

$V_{100} = 0.1\theta_{100} + 0.9(0.1\theta_{99} + 0.9(0.1\theta_{98} + 0.9(\cdots)))$

$= 0.1\theta_{100} + 0.1\times0.9\cdot\theta_{99} + 0.1(0.9)^2\theta_{98} + 0.1(0.9)^3\theta_{97} + 0.1(0.9)^4\theta_{96} + \cdots$

$0.9^{10} \approx 0.35 \approx \frac{1}{e}$

$(1-\varepsilon)^{1/\varepsilon} = \frac{1}{e}$

$\varepsilon = 0.02 \rightarrow 0.98^{50} \approx \frac{1}{e}$

Andrew Ng

$0.1\theta_{98} + 0.9v_{97}$

$0.98^?$

$V_2 = \beta v_1 + (1-\beta)\theta_2$

$V_1 = \beta v_0 + (1-\beta)\theta_1$

$v_0 = \theta_0$

原书: 0.1

∴ 近似平均了 $\frac{1}{1-\beta}$ 天的数据.

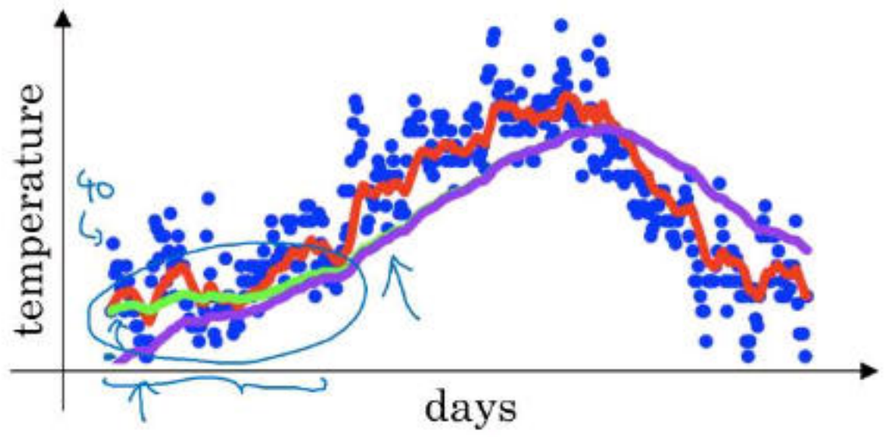
指数 Decay.

代码只需维护变量 V.

问题:

日期: 12 / 17

Bias correction



$\beta = 0.98$

$\rightarrow v_t = \beta v_{t-1} + (1-\beta)\theta_t$

$v_0 = 0$

$v_1 = 0.98 v_0 + 0.02\theta_1$

$v_2 = 0.98 v_1 + 0.02\theta_2$

$= 0.98 \times 0.02 \times \theta_1 + 0.02\theta_2$

$= 0.0196\theta_1 + 0.02\theta_2$

$\frac{v_t}{1-\beta^t}$

$t=2: \quad 1-\beta^t = 1-(0.98)^2 = 0.0396$

$\frac{v_2}{0.0396} = \frac{0.0196\theta_1 + 0.02\theta_2}{0.0396}$

Andrew Ng

$v_0 = 0$.

问题: $v_0 = 0$ $v_1 = 0.98 v_0 + 0.02\theta_1 = 0.02 \times 40$

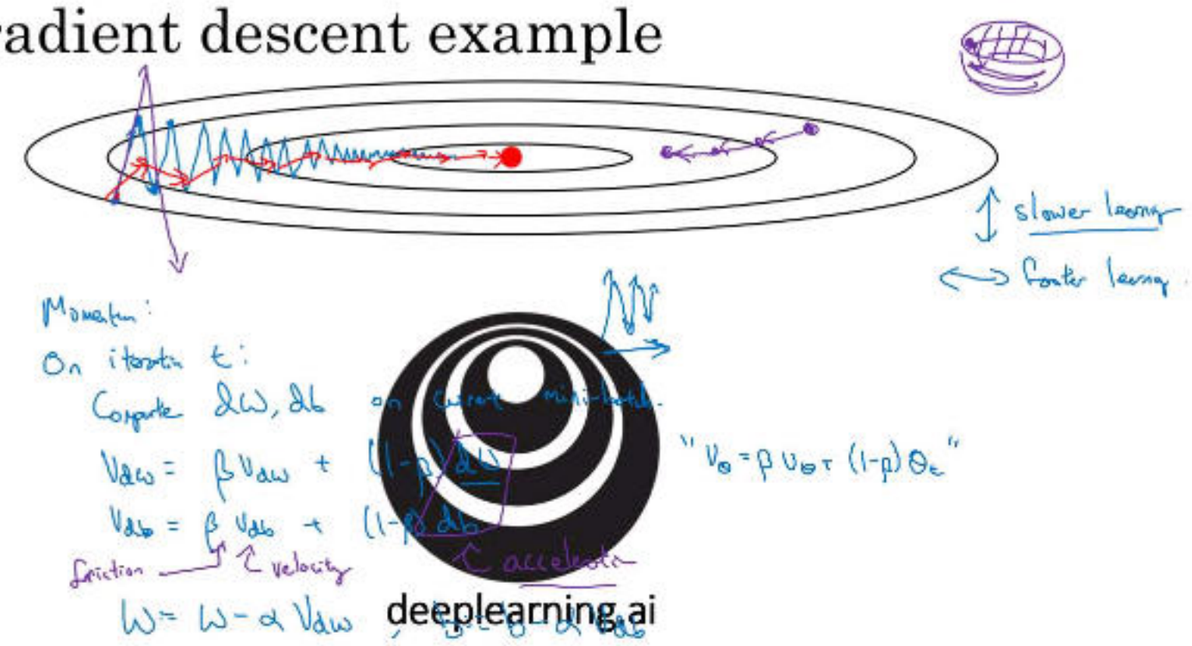
$v_2 = 0.98 v_1 + 0.02\theta_2 = 0.98 \times 0.02\theta_1 + 0.02\theta_2$. Bias.

$\frac{v_t}{1-\beta^t}$ $\beta \to 1, \ \beta^t \to 0$ $(t \uparrow)$.

日期: 12 / 17

三种优化算法:

Gradient descent example



deeplearning.ai

Andrew Ng

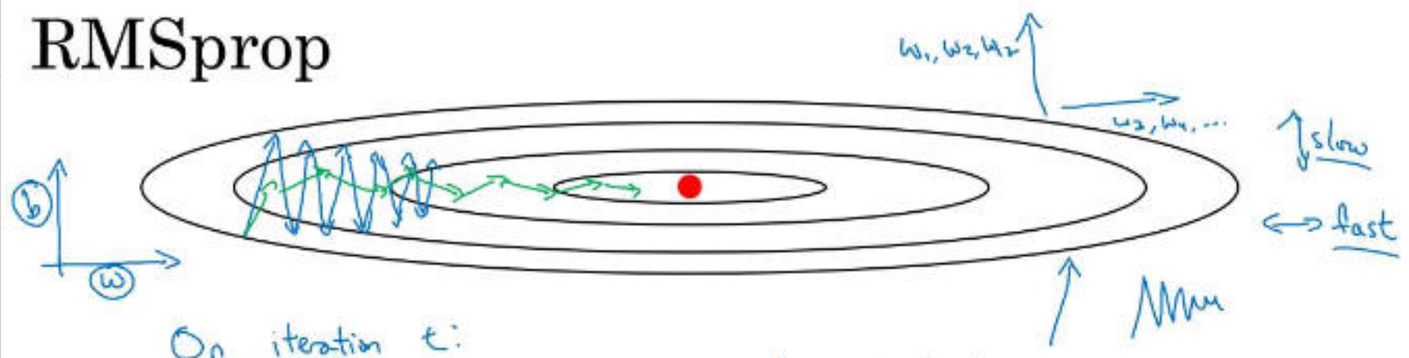
a): Momentum:

$V_{dw} = \beta V_{dw} + (1-\beta) dw$. 对 dw, db 做

$V_{db} = \beta V_{db} + (1-\beta) db$. 指数加权平均.

垂直方向抵消, 减少无谓的摆动.

RMSprop



On iteration t:

Compute dW, db on current mini-batch

$S_{dW} = \beta_2 S_{dW} + (1-\beta_2) dW^2$ ← element-wise ← small

$S_{db} = \beta_2 S_{db} + (1-\beta_2) db^2$ ← large

$W := W - \alpha \frac{dW}{\sqrt{S_{dW}+\varepsilon}}$ ← $b := b - \alpha \frac{db}{\sqrt{S_{db}+\varepsilon}}$ ←

$\varepsilon = 10^{-8}$

Andrew Ng

b): Root Mean Square prop.

$S_{dw} = \beta_2 S_{dw} + (1-\beta_2) dw^{[2]}$ → element-wise.

$S_{db} = \beta_2 S_{db} + (1-\beta_2) db^{[2]}$.

$W = W - \alpha \cdot \frac{dw}{\sqrt{S_{dw}+\varepsilon}}$ ↙ $b = b - \frac{db}{\sqrt{S_{db}+\varepsilon}}$ ↗.

dw ↑, S_{dw} ↑.

S_{db} ↓.

日期: 12 / 17

Adam optimization algorithm

$V_{dw}=0$, $S_{dw}=0$. $V_{db}=0$, $S_{db}=0$

On iteration t:

Compute dW, db using current mini-batch

$V_{dw} = \beta_1 V_{dw} + (1-\beta_1) dW$, $V_{db} = \beta_1 V_{db} + (1-\beta_1) db$ ← "momentum" β_1

$S_{dw} = \beta_2 S_{dw} + (1-\beta_2) dW^2$, $S_{db} = \beta_2 S_{db} + (1-\beta_2) db$ ← "RMSprop" β_2

yhat = np.array([.9, 0.2, 0.1, .4, .9])

$V_{dw}^{corrected} = V_{dw} / (1-\beta_1^t)$, $V_{db}^{corrected} = V_{db} / (1-\beta_1^t)$

$S_{dw}^{corrected} = S_{dw} / (1-\beta_2^t)$, $S_{db}^{corrected} = S_{db} / (1-\beta_2^t)$

$$W := W - \alpha \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}} + \varepsilon} \qquad b := b - \alpha \frac{V_{db}^{corrected}}{\sqrt{S_{db}^{corrected}} + \varepsilon}$$

Andrew Ng

c): Adaptive momentum.

对 dw、db 指数加权.

同时对修正后的 V_{dw}、V_{db} 做 "RMS".

两种优化方法的叠加.